RECEIVED
CENTRAL FAX CENTER

AUG 31 2006

Docket No. AT9-98-441 ***PATENT***

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Chao et al.**

Serial No.: **09/282,907** Group Art Unit: **2172**

Filed: **March 31, 1999** Examiner: **Fleurantin, Jean B.**

For: **Error Detection Protocol**

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on August 31, 2006.

By: Dell Whitton
Dell Whitton

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF (37 C.F.R. 41.41)

This Reply Brief is submitted in response to the Examiner's Answer mailed on July 16, 2002 and to the Notification of Non-Compliant Appeal Brief mailed on December 1, 2005.

No fees are believed to be required to file a Reply Brief. If any fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447.

REMARKS

I. RESPONSE TO NOTIFICATION OF NON COMPLIANT REPLY BRIEF

The examiner notified that the reply brief filed on August 20, 2002 did not contain a correct copy of appealed claims. A correct copy of appealed claims is attached as an appendix to the present reply brief.

The examiner notified that the petition for extension of time was deficient in fees required. Authorization to charge the required fees to IBM Corporation's deposit account is attached herewith.

The examiner notified that the appendix to the appeal brief filed April 15, 2002 did not reflect changes to claim 25 as indicated in a prior approved amendment. The copy of appealed claims appended to the present reply brief includes a correct copy of the amended claim 25.

II. RESPONSE TO EXAMINER'S RESPONSE TO APPELLANTS' ARGUMENTS

A. San Andres does not teach or suggest "each node in the computer cluster voting based on a functional outcome of the database update request" as recited in claim 1.

In the Examiner's Answer, the Examiner states that "San Andres includes the steps of each time an update transaction is dispatched by the arbiter the arbiter monitors the outcome 'success or failure' of the transaction on each server by checking the status codes returned by the server, which is readable as detecting an out-of-sync condition as a result of a different functional outcome, see col. 19, lines 43-46. Thus, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teaching of San Andres with the step of each node in the computer cluster voting based on a functional outcome of the database update request. This modification would allow the teachings of San Andres to improve the accuracy of the error detection protocol." Examiner's Answer, Page 11.

The Examiner has not provided a proper motivation for modifying San Andres. The Examiner must submit objective evidence in support of modifying San Andres. *In re Lee,* 61 USPQ2d 1430, 1433 (Fed. Cir. 2002); *In re Kotzab, 55* USPQ2d 1313, 1316-13 17 (Fed. Cir. 2000). Instead, the Examiner simply relies on the Examiner's own subjective opinion as support for modifying San Andres which is insufficient. *Id.* The Examiner simply states that by modifying San Andres with the above limitation that it improves the accuracy of the "error detection protocol" which happens to be the title of application. No where does San Andres discuss or use the terms "error detection protocol." This is an improper motivation provided by the Examiner. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 1 in view of the cited prior art.

B. San Andres does not teach or suggest *"refreshing* the database *in response to the detecting step"* as recited in claim 3.

The Examiner states that "because San Andres includes the step of when an application server of a service group receives a client request that indicates a modification to replicated service content data the server generates an update transaction and sends the update transaction to the arbiter, which is readable as refreshing the database in response to the detecting." Examiner's Answer, Page 11. As interpreted by the Appellants, San Andres simply teaches that when the application server receives a request to modify the replicated service content data, the application server generates an update transaction. *However, the application server does not generate the update transaction in response to detecting an out-of-sync condition as a result of a different functional outcome.* The application server simply receives a request to modify the replicated service content data from a client. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 3 in view of the cited prior art.

C. San Andres does not teach or suggest that "the *out-of-sync condition is an error"* as recited in claim 2.

The Examiner states that "San Andres includes steps of the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being

inconsistent with final outcome and taken off line; which is readable as the out-of-sync condition is an error." Examiner's Answer, Pages 11-12. As interpreted by the Appellants, San Andres teaches that an Arbiter may use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance. As interpreted by the Appellants, San Andres simply teaches an Arbiter using a scheme to determine which servers are taken off-line but does not teach detecting an out-of-sync condition that is an error. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 2 in view of the cited prior art.

D. San Andres does not teach or suggest *"voting, by all of the other nodes in the computer cluster* to approve update if a match results from the comparison" as recited in claims 9 and 21 and similarly in claim *15.*

The Examiner states that "San Andres includes the step that whenever the arbiter replicates a transaction the arbiter monitors the outcome of the transaction on each server of the service group to ensure consistent processing of the transaction by all such servers, when one or more servers indicates a different outcome than the other servers of the service group the arbiter uses a voting scheme to resolve the conflict between the servers a preferred voting scheme is described below under the heading status codes and conflict resolution; which is read as voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison; see col. 7, lines 10-19." Examiner's Answer, Page 12. San Andres further teaches that 'when one or more servers 120 indicates a different outcome than the other servers of the service group, the Arbiter uses a voting scheme to resolve the conflict between the servers." Column 17, Lines 14-17. As interpreted by the Appellants, San Andres simply teaches an ***Arbiter*** that may ***use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off line for maintenance.*** Consequently, San Andres does not teach or suggest ***voting by all of the other nodes in the computer cluster*** to approve update if a match results from the comparison. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

E. San Andres does not teach or suggest *"comparing, by all of the other nodes* in the computer cluster, the *update results to results of application of the update to the local copy of the database"* as recited in claims 9 and 21 and similarly in claim 15.

The Examiner states that "San Andres includes the steps of the arbiter replicates service content data by dispatching atomic transactions 'which are generated by the arbitered services' to groups servers 120 these transactions are in the form of update commands (referred to herein as 'update transactions') which when interpreted by the receiving service applications typically specify an update to a specific data entity or set of data entities each server 120 which receives the update transaction from the arbiter processes the transaction, which normally involves updating the serves locally-stored service content data; which is readable as comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; see col. 16, lines 50." Examiner's Answer, Page 13. As interpreted by the Appellants, San Andres teaches that a server may receive an update transaction from the Arbiter to update the server's locally-stored service content data. However, San Andres does not teach that *all of the other nodes in the computer cluster compare* the update results to results of application of the update to the local copy of the database. Furthermore, San Andres does not teach that all of the other nodes in the computer cluster compare the *update results to results of application of the update to the local copy of the database.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

F. San Andres does not teach or suggest *"broadcasting an approval* of the *update to the database if all of the other nodes vote to approve the update"* as recited in claims 11, 22 and similarly in claim 17.

The Examiner states that "San Andres includes the steps of when a new application server is brought on line previously dispatched update transactions stored in the transaction log are dispatched in sequence to the new server to bring the new server's content data up to date, which is readable as broadcasting an approval of the update to the database if all of the other nodes vote to approve the update; see abstract, lines 16-20." Examiner's Answer, Page 13. As

interpreted by the Appellants, San Andres teaches dispatching update transactions to the new server to bring the new server's content data up to date. However, San Andres does not teach or suggest *broadcasting an approval* of the *update to the database*. Furthermore, San Andres does not teach or suggest broadcasting an approval of the update to the database *if all of the other nodes vote to approve the update*. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 11, 17 and 22 in view of the cited prior art.

G. Other Matters Raised by the Examiner

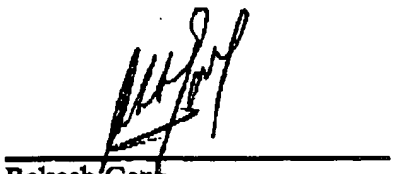
All other matters raised by the Examiner, e.g., Examiner's reliance on *In re McLaughlin*, Examiner's assertion to give claim limitations their broadest reasonable interpretation, have been adequately addressed in Appellant's Appeal Brief and therefore will not be addressed herein for the sake of brevity.

H. Comments

It is noted that words are italicized only for emphasis. Words that are italicized are not meant to imply that only those words are not taught or suggested in the cited prior art.

III. CONCLUSION

For the reasons stated in Appellants' Appeal Brief and noted above, Appellants respectfully assert that the rejection of claims 1-25 is in error. Appellants respectfully request reversal of the rejections and allowance of claims 1-25.



Rakesh Garg
Reg. No. 57,434
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, Texas 75230
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method for maintaining a consistent set of replicas of a database within a computer cluster, comprising the steps of:

each node in the computer cluster receiving a database update request;

each node in the computer cluster voting based on a functional outcome of the database update request; and

detecting an out-of-sync condition as a result of a different functional outcome.

2. The method as recited in claim 1, wherein the out-of-sync condition is an error.

3. The method as recited in claim 1, further comprising the step of:

refreshing the database in response to the detecting step.

4. The method as recited in claim 1, further comprising the step of:

resetting cluster membership in response to the detecting step.

5. The method as recited in claim 1, further comprising the step of:

blocking further participation by the node having the out-of-sync condition in response to the detecting step.

6. The method as recited in claim 1, further comprising the step of:

declaring an end-of-transaction state on update voting completion when the database update is being done in a transactional manner.

7. The method as recited in claim 6, further comprising the step of:

backing out an update when update voting does not meet a criteria established for success.

8. The method as recited in claim 7, wherein the criteria established for success is that no more than one node has inconsistent results.

9. A method for maintaining a consistent set of replicas of a database within a computer cluster, comprising the steps of:

broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

node requesting update broadcasts results of update to all of the other nodes in the computer cluster;

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; and

voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison.

10. The method as recited in claim 9, further comprising the step of:

voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison.

11. The method as recited in claim 9, further comprising the step of:

broadcasting an approval of the update to the database if all of the other nodes vote to approve the update.

12. The method as recited in claim 10, further comprising the step of:

if more than one of the plurality of nodes votes to continue, performing a recovery process.

13. The method as recited in claim 12, wherein the recovery process further comprises the step of:

if more than a specified number of the nodes voted to continue, backing out the update to the database.

14. The method as recited in claim 12, wherein the recovery process further comprises the step of:

if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes.

15. A computer cluster operable for maintaining a consistent set of replicas of a database within the computer cluster, comprising:

a group services client operable for broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

the plurality of nodes coupled to the computer cluster operable for applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

circuitry for broadcasting results of the update to all of the other nodes in the computer cluster;

circuitry for comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; and

circuitry for voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison.

16. The computer cluster as recited in claim 15, further comprising:

circuitry for voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison.

17. The computer cluster as recited in claim 15, further comprising:

circuitry for broadcasting an approval of the update to the database if all of the other nodes vote to approve the update.

18. The computer cluster as recited in claim 16, further comprising:

if more than one of the plurality of nodes votes to continue, circuitry for performing a recovery process.

19. The computer cluster as recited in claim 18, wherein the recovery process further comprises:

if more than a specified number of the nodes voted to continue, circuitry for backing out the update to the database.

20. The computer cluster as recited in claim 18, wherein the recovery process further comprises:

if less than a specified number of the nodes voted to continue, circuitry for performing the recovery process on the specified number of the nodes.

21. A computer program product adaptable for storage on a computer readable medium, the computer program product operable for maintaining a consistent set of replicas of a database within a computer cluster, comprising the program steps of:

broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

node requesting update broadcasts results of update to all of the other nodes in the computer cluster;

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database;

voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison; and

voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison.

22. The computer program product as recited in claim 21, further comprising the program step of:

broadcasting an approval of the update to the database if all of the other nodes vote to approve the update.

23. The computer program product as recited in claim 22, further comprising the program step of:

if more than one of the plurality of nodes votes to continue, performing a recovery process.

24. The computer program product as recited in claim 23, wherein the recovery process further comprises the program step of:

if more than a specified number of the nodes voted to continue, backing out the update to the database.

25. The computer program product as recited in claim 24, wherein the recovery process further comprises the program step of:

if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes.